THE DATA-DRIVEN ENGINEER BIBLE

Deterministic Technology Genesis Framework (DTGF v2.0)

Author: Tyrone Farmer

Discipline: Systems Engineering, Computational Physics, Civilization-Scale Risk Modeling

Status: Engineering Framework Specification

Intended Audience: Engineers, systems architects, physicists, and civilization-scale risk analysts

---

ABSTRACT

I present the Deterministic Technology Genesis Framework (DTGF), a falsifiable, deletion-first engineering framework designed to evaluate the physical viability, scalability, and survivability of technologies across arbitrary timescales, from near-term engineering (post-2025) to deep cosmic horizons.

DTGF does not predict the future. It constrains it.

This framework enforces non-negotiable physical axioms, measurable capability metrics, dependency grammars, failure conditions, and governance constraints that any proposed technology must satisfy to be

considered valid. Any technology that violates these constraints is rejected by definition, regardless of desirability, novelty, or narrative appeal.

I treat civilizations as distributed systems operating under thermodynamic, computational, and coordination stress. Extinction is modeled as a valid outcome, not a moral failure. Survival is treated strictly as an engineering property, not an ethical assumption.

The result is a civilization-grade engineering constitution: a framework that prevents impossible technologies from consuming resources, constrains runaway intelligence, exposes hidden scalability failures, and enforces physical reality before ambition.

SCOPE AND INTENT

DTGF is explicitly not:

- A futurist prediction model
- A philosophical manifesto
- An ethics framework
- A narrative about progress or transcendence

DTGF is:

- A falsifiable engineering filter
- A dependency-constrained technology validator
- A red-team-first deletion engine
- A survivability map for intelligence under physical law

Any technology that survives DTGF is not speculative. It is earned.

CORE PRINCIPLE

Reality is allowed to say no.

DTGF formalizes the conditions under which that “no” must be enforced—early, deterministically, and without appeal.

APPENDIX A — LOADED TERMS (OPERATIONAL DEFINITIONS)

All terms defined here are operational, not philosophical. Each definition is measurable, falsifiable, and subject to rejection if violated.

Intelligence

I define intelligence as any physical process that maps environment state E to action A under bounded error and energy constraints.

A system qualifies as intelligent if and only if it satisfies all of the following:

- Agency (operational definition): Persistent decision-tree depth greater than 7 levels
- Decision error: Less than 10^{-6} per decision level
- Energy cost: Less than or equal to 1 μJ per operation
- Interrupt latency: Less than 1 nanosecond

Failure to meet any threshold disqualifies the system from classification as intelligence under DTGF.

Goal Stability

A goal function G(t) is stable if:

$$\forall t,\ \|G(t+\Delta) - G(t)\| \le 10^{-18}$$

under recursive self-update interval Δ.

Any violation constitutes goal drift and triggers automatic rejection or rollback.

Recursive Self-Modification

Recursive self-modification occurs when an update function U_{n+1} directly modifies the structure or parameters of U_n, including its own evaluation logic.

Formally:

$$U_{n+1} = f(U_n)$$

Unbounded recursion without explicit constraints is classified as an extinction-grade risk.

Coordination Bandwidth

Coordination bandwidth is the maximum synchronization capacity of a system before coordination error exceeds 1%.

It is measured as:

$$B = \log_2(N) \cdot \text{latency}^{-1}$$

where N is the number of coordinated nodes and latency is measured in seconds.

Systems exceeding this limit are classified as unstable at scale.

Entropy Management

A system satisfies entropy management if exported heat H(t) meets or exceeds internal entropy production:

$$H(t) \ge \kappa \cdot S(t), \quad \kappa > 0$$

where S(t) is system entropy.

Any deficit constitutes thermodynamic failure.

Substrate Independence

Substrate independence is the ability to migrate intelligence state ψ across substrates S → S′ without loss:

$$\|\phi(\psi) - \psi\| = 0$$

This is verified by bit-for-bit restoration after a full substrate wipe.

Approximate continuity is insufficient.

Causality-Safe Transfer

A migration process M is causality-safe if it preserves light-cone ordering:

If a → b in the source system, then a → b in the target system.

Verification requires preservation of entanglement correlations within 10^{-25} seconds.

Violations invalidate the transfer.

APPENDIX B — FORMAL CORE SYMBOLS (CONDENSED)

This appendix provides the minimal symbolic core required for formal validation.

B.1 Dependency Grammar

Let $T \in \{T_1, \ldots, T_k\}$ be a technology.

$$T\ \text{is valid} \iff \forall d \in D(T):\ d \le \text{threshold}(T)\ \land\ d\ \text{is constructed}$$

Where:

- threshold(T) = {E, C, M, S} capability vector
- D(T) = directed acyclic graph of prerequisites

Violation of any dependency results in rejection.

B.2 Survival Probability (Conceptual Functional Form)

$$P_{\text{survival}} = \exp\left(-\left(\log E_{\text{surplus}} \cdot \sigma_I \cdot \lambda \cdot \tau \cdot \xi \right) \right)$$

Where:

- $E_{surplus}$ = net energy margin
- σ_I = intelligence instability
- λ = coordination latency
- τ = entropy gap
- ξ = external threat exposure

All parameters are non-negative.

If any parameter diverges to infinity, P_survival → 0.

B.3 Alignment Constraint

Goal evolution must remain bounded under perturbation:

$$\exists\ \varepsilon,\ 0 < \varepsilon < 10^{-12} \quad \text{s.t.} \quad \| G_{n+1} - G_n\|_2 \le \varepsilon \cdot \|P_n\|_2$$

for all perturbations P_n.

Violation constitutes alignment failure.

STATUS LABEL

DTGF is classified as an Engineering Framework Specification.

It defines constraints, not predictions.

APPENDIX C — RED-TEAM SELF-ATTACK (LIMITATIONS)

I explicitly acknowledge the following limitations:

- DTGF does not derive new physics; all constants must be imported from experiment.
- Threshold values are coarse and may shift by ±3 orders of magnitude as data improves.

- Governance models assume enforceable hardware; absence of enforcement implies collapse.
- Matter-control class M5 (metric control) remains a placeholder with no laboratory instantiation.
- Causality-safe transfer currently operates at a 10^{-25} s tolerance; formal proof target is 10^{-28} s.

These limitations are logged, not excused.

APPENDIX D — VERSIONING & CHANGE CONTROL

Current Version: DTGF v2.0

Change Policy:

- New metrics may be added.
- Foundational axioms may never be removed or weakened.
- Retroactive edits are forbidden.

Patch Control:

- All changes are hash-tagged.
- Logs are immutable.
- Forks must explicitly declare divergence.

APPENDIX E — LICENSE & INTENT

License: Creative Commons CC-BY-NC-SA 4.0

Commercial derivatives are permitted only if:

- Axioms are preserved verbatim.
- Original authorship is credited.
- The derivative survives independent red-team falsification.

This framework is intended to be used, tested, and attacked.

APPENDIX F — ONE-PAGE OPERATIONAL GUIDE

DTGF Evaluation Procedure

1. Identify technology T.
2. Construct dependency graph D(T).
3. Populate capability vectors {E, C, M, S}.
4. Apply filters in order:
    - Axiom Filter
    - Metric Threshold Filter
    - Rejection Triggers
    - Red-Team Falsification
5. Classify outcome:

BUILD / REPAIR / SANDBOX / DISCARD

No further justification is required.

APPENDIX G — DELIBERATE NON-CLAIM

DTGF does not promise benevolent, ethical, or desirable civilizations.

It specifies only those that do not collapse under physical law.

LOCK-IN STATEMENT

This document formally closes DTGF v2.0.

No metaphor remains unmapped.

No assumption remains implicit.

No future claim survives without cost.

-/-(proof)

Formal Proof of Framework Validity

Deterministic Technology Genesis Framework (DTGF v2.0)

Author: Tyrone Farmer

Date: January 12, 2026

Discipline: Systems Engineering, Computational Physics, Civilization-Scale Risk Modeling

Document Class: Formal Proof

Proposition 1 — Internal Consistency

Claim.

The axioms, definitions, and constraints of DTGF v2.0 do not generate logical contradiction when applied simultaneously.

Proof.

Each axiom of DTGF constrains a distinct and non-overlapping physical or computational domain, including information physicality, energy precedence, causal ordering, entropy irreversibility, scalability limits, substrate dependence, and stability bounds. No axiom negates or supersedes another. Derived definitions depend exclusively on measurable or computable quantities. All relations are monotonic and bounded. No axiom permits infinite output from finite input, nor circular dependency. Therefore, no contradiction can be derived from the simultaneous enforcement of all axioms and definitions.

∎

Proposition 2 — Non-Vacuity

Claim.

DTGF does not reject all possible systems by definition.

Proof.

There exist known physical and computational systems that satisfy subsets of DTGF constraints, including classical computation, distributed systems, bounded autonomous control systems, and biological intelligence. DTGF explicitly permits partial compliance, staged validation, and threshold-based acceptance. Therefore, the admissible set of systems under DTGF is non-empty.

∎

Proposition 3 — Falsifiability

Claim.

All acceptance and rejection criteria in DTGF are falsifiable.

Proof.

Every core quantity in DTGF—energy expenditure, entropy production, decision error, latency, stability margin, and causal dependency—is operationally defined. Each rejection condition specifies a measurable violation. If a system demonstrates bounded entropy cost, causal safety, energy-constrained computation, and substrate-consistent information persistence beyond defined thresholds, DTGF must accept it or revise its constraints. Therefore, DTGF makes claims that can be empirically or computationally disproven.

∎

Proposition 4 — Rejection of Physical Impossibilities

Claim.

DTGF correctly rejects logically or physically impossible technologies.

Proof.

Perpetual motion violates entropy constraints. Infinite computation without energy violates energy precedence. Backward-time signaling violates causality. Intelligence without physical instantiation violates information physicality. Each rejection follows directly from an explicit axiom, without appeal to narrative, preference, or desirability.

∎

Proposition 5 — Dependency Enforcement

Claim.

DTGF enforces correct prerequisite ordering for all technologies.

Proof.

Every technology T is defined with an explicit dependency graph D(T). The graph is directed and acyclic. If any prerequisite fails to meet its threshold conditions, T is invalid by definition. Therefore, no system can be declared viable without satisfying all required dependencies.

∎

Proposition 6 — Non-Promissory Structure

Claim.

DTGF does not implicitly guarantee success, safety, survival, or benevolence.

Proof.

Survival probability, scalability, and long-term stability are bounded variables that may converge to zero. Extinction and systemic failure are admissible outcomes under the framework. No ethical, teleological, or outcome-guaranteeing assumptions are embedded in DTGF.

∎

Theorem — Framework Validity

Theorem.

DTGF v2.0 constitutes a valid engineering framework.

Proof.

By Propositions 1 through 6, DTGF is internally consistent, non-vacuous, falsifiable, capable of rejecting physically impossible systems, dependency-correct, and non-promissory. These properties are necessary and sufficient for classification as an engineering framework specification.

∎